

Linear Algebra for Computational Engineering

Stephen Wood

A library that provides accurate answers, effectively utilizes modern architectures, enable solver development, facilitates software maintenance, and reduces the time to solution for finite element discretizations is not easy to find. The outline of a linear algebra library designed for integration in the workflows of finite element software is presented. The convergence of the Flexible GMRES Method variants implemented in LACE and the FGMRES implementation within Intel’s 2017 Math Kernel Library for a benchmark problem from the field of computational fluid dynamics are also presented.

1 Motivation

To wholly harness current and emerging high performance computer architectures CFD software needs to utilize multilevel decomposition across nodes with fine grained parallelism on threads of a CPU or accelerator. In particular the residual calculation needs graph coloring per subdomain for effective multi-threading. Subtle variations in work sharing patterns have consequences for numerics, algorithmic performance, and time to solution. A resource of linear algebra components tailored for modern architectures and extensible to emerging architectures will be useful for the development and modernization of finite element software. Such a library will be a contribution to the field of computational fluid dynamics. This is the goal of the Linear Algebra for Computational Engineering (LACE) Library.

2 LACE Library Overview

An outline of the LACE library is shown in Figure 1. Many of the elements shown in Figure 1 have been implemented. Those elements that are planned for future releases are denoted with (future) on the outline.

As an example of a specific implementation, a more detailed view of the Generalized Minimal Residual Method within LACE is shown in Figure 2.

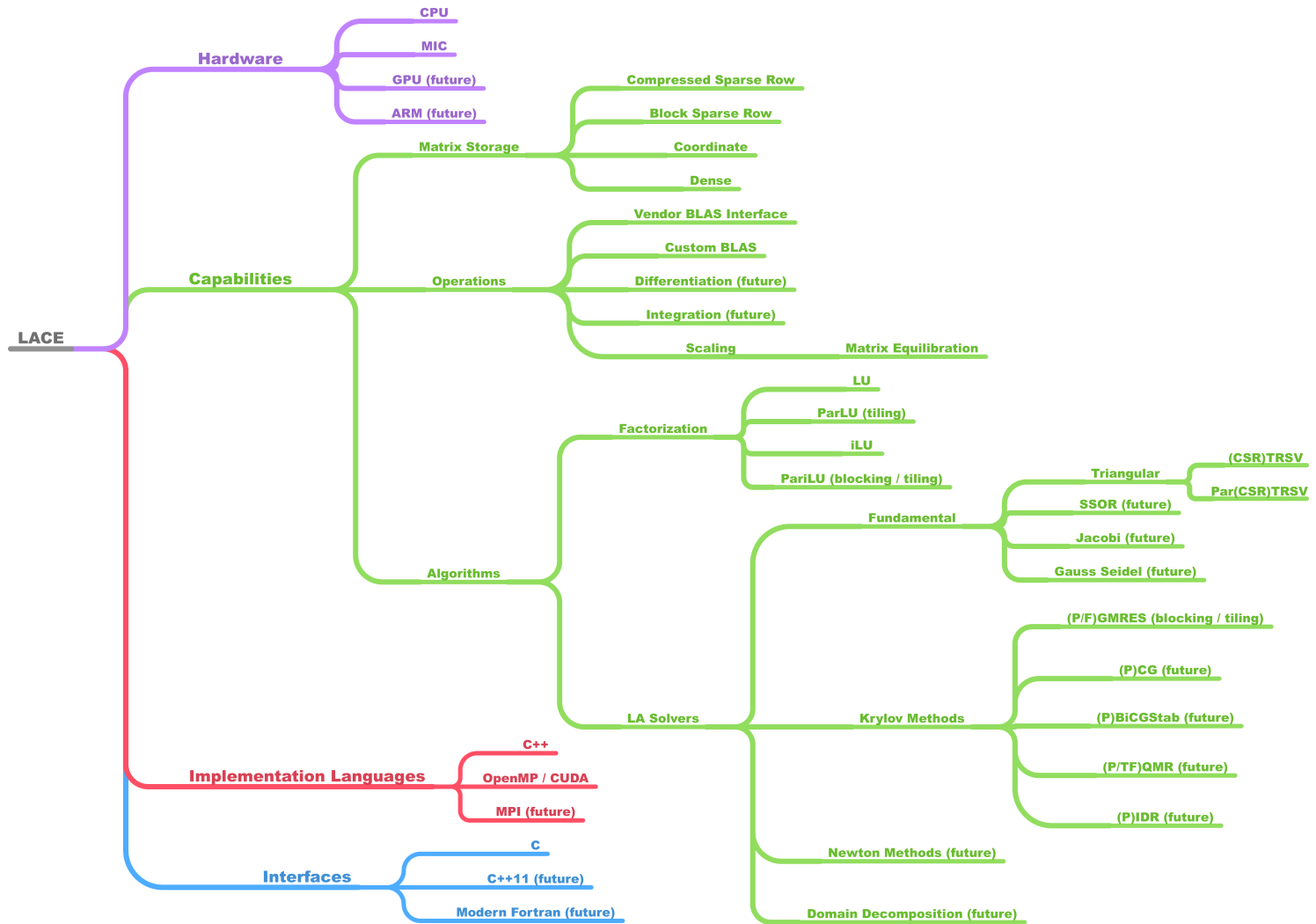


Figure 1: Outline of significant features of LACE that have been realized or are planned for future release.

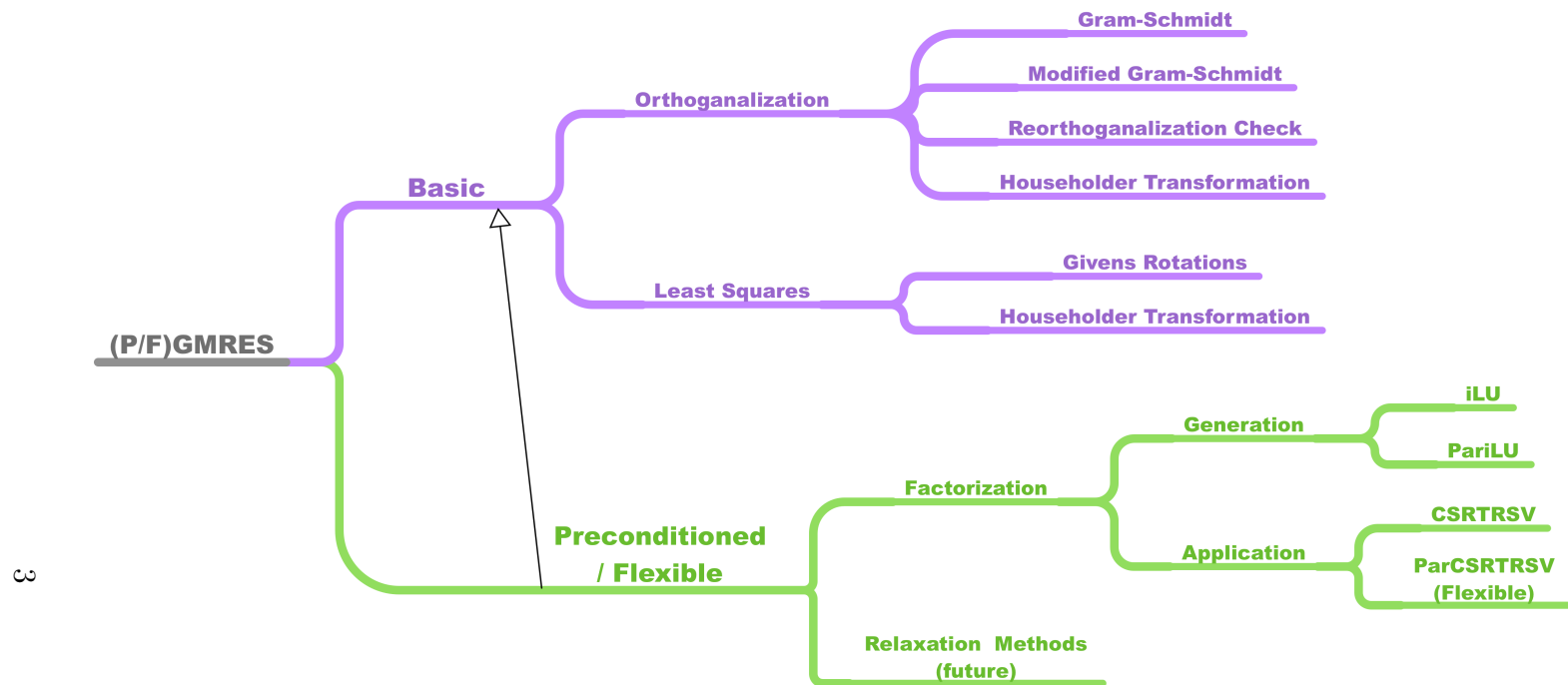


Figure 2: Outline of Generalized Minimal Residual Method implementation within LACE.

3 Test Problem

Variants of the GMRES solver implemented within LACE have been applied to the 30p30n multi-component airfoil benchmark problem from the BANC-II workshop to examine the impact of preconditioner generation and application in serial and parallel. The domain discretization is shown in Figure 3. The solution and matrix are visualized in Figure 4.

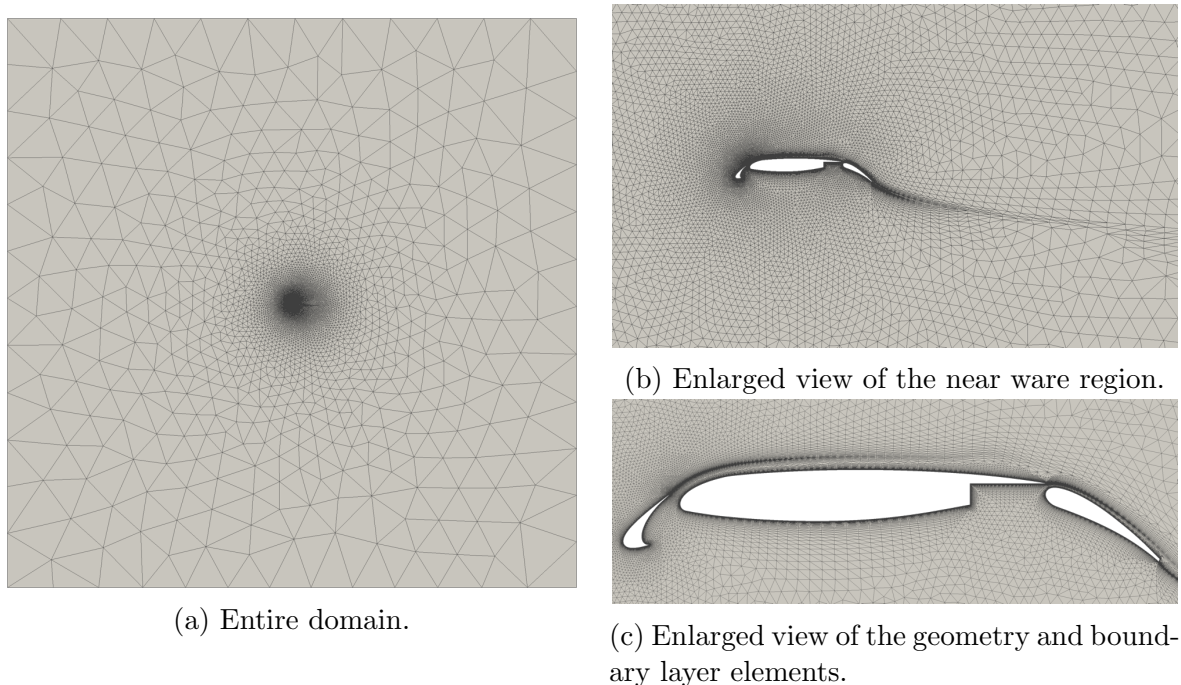
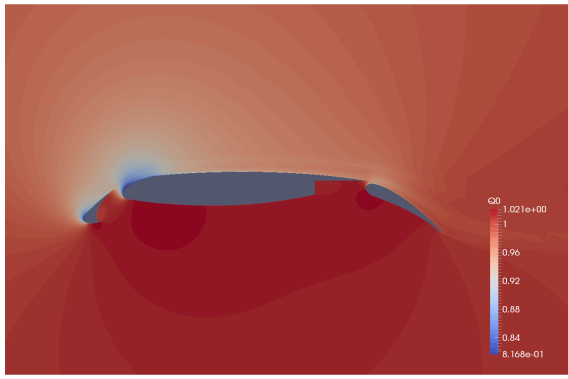


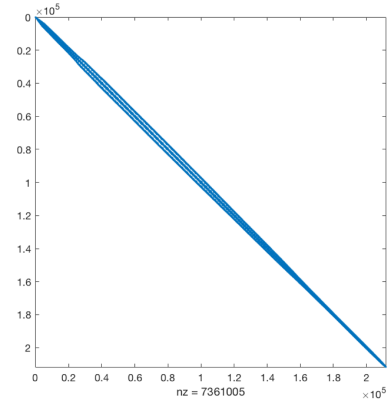
Figure 3: Domain discretization for the 30p30n multi-component airfoil benchmark problem.

The results shown in Figure 5 were generated with a MacBook Pro (Mid 2014) while the Beacon cluster at the JICS Advanced Computing Facility was undergoing maintenance. Eight OpenMP threads were utilized for preconditioner generation by the Parallel incomplete LU (PariLU) factorization algorithm. Eight OpenMP threads were also utilized for preconditioner application by the Parallel Compressed Sparse Row Triangular Solves (ParCSRTRSV) algorithms.

It should be noted that orthogonalization was carried out with the Modified Gram-Schmidt algorithm and Givens Rotations were applied to solve the least squares problem within the Flexible GMRES



(a) Visualization of non-dimensionalized density field.



(b) Sparsity pattern of coefficient matrix for the simulation.

Figure 4: Visualization of the flow around the multi-component airfoil in physical space in juxtaposition with the matrix of coefficients.

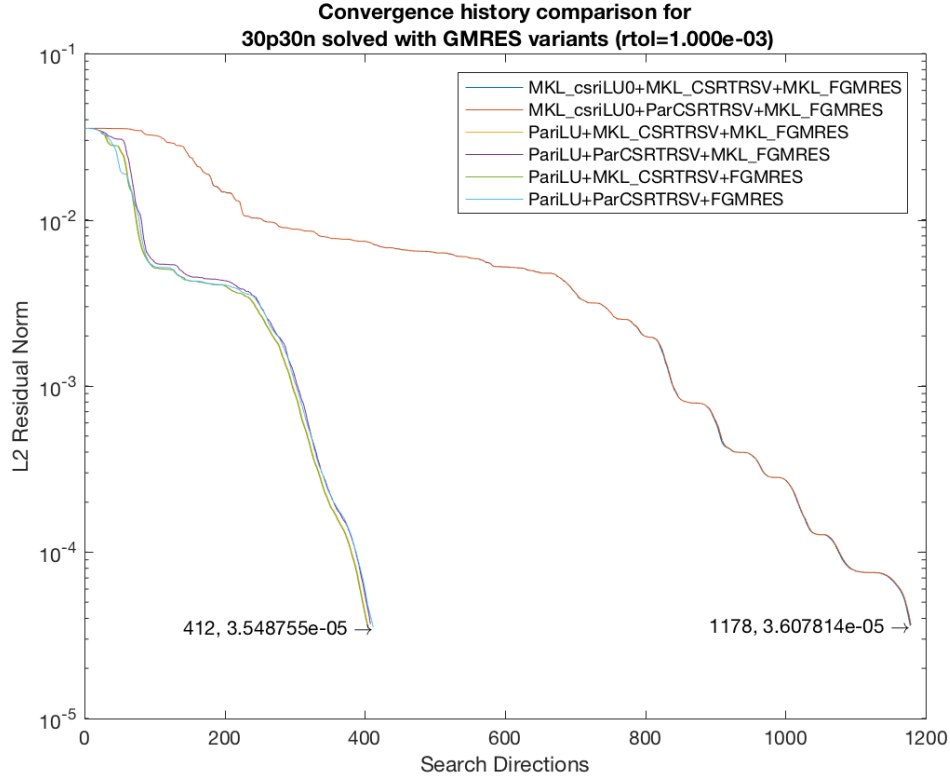


Figure 5: Comparison of convergence history for a sampling of Flexible GMRES variants implemented to date within LACE and the Intel MKL 2017 FGMRES Method.